Scalable Multi-Robot Informative Path Planning for Target Mapping via Deep Reinforcement Learning

Apoorva Vashisth[1], Manav Kulshrestha[1], Damon Conover[2], Aniket Bera[1]
[1]*Department of Computer Science, Purdue University, USA* [2]*DEVCOM Army Research Laboratory, USA*
{vashista, mkulshre, aniketbera}@purdue.edu, damon.m.conover.civ@army.mil

***Abstract*—Autonomous robots are widely utilized for mapping and exploration tasks due to their cost-effectiveness. Multi-robot systems offer scalability and efficiency, especially in terms of the number of robots deployed in more complex environments. These tasks belong to the set of Multi-Robot Informative Path Planning (MRIPP) problems. In this paper, we propose a deep reinforcement learning approach for the MRIPP problem. We aim to maximize the number of discovered stationary targets in an unknown 3D environment while operating under resource constraints (such as path length). Here, each robot aims to maximize discovered targets, avoid unknown static obstacles, and prevent inter-robot collisions while operating under communication and resource constraints. We utilize the centralized training and decentralized execution paradigm to train a single policy neural network. A key aspect of our approach is our coordination graph that prioritizes visiting regions not yet explored by other robots. Our learned policy can be copied onto any number of robots for deployment in more complex environments not seen during training. Our approach outperforms state-of-the-art approaches by at least 26.2% in terms of the number of discovered targets while requiring a planning time of less than 2 sec per step. We present results for more complex environments with up to 64 robots and compare success rates against baseline planners[1]**

***Index Terms*—Motion and Path Planning; Reinforcement Learning; Multi-Robot Systems**

I. Introduction

AUTONOMOUS robotic systems are used in several tasks, such as search and rescue missions [1], environment mapping [2], and orchard monitoring [3]. Multi-robot systems are gaining popularity in these domains due to their increased efficiency, compared to single-robot systems [4] and manual approaches [5]. Key challenges for deploying multi-robot systems in these tasks include planning efficient paths for all robots to optimize the task objective, avoiding inter-robot and robot-obstacle collisions, scaling to larger multi-robot systems deployed in more complex environments, and considering communication and resource constraints.

In this work, we aim to develop a deep reinforcement learning-based, scalable, multi-robot path planning approach for discovering stationary targets in an unknown 3D environment. Here, each robot is constrained to a resource

Manuscript received: June, 2, 2025; Revised November, 6, 2025; Accepted December, 16, 2025.

This paper was recommended for publication by Editor Dr. Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the DEVCOM Army Research Laboratory under cooperative agreement W911NF2520170.

Digital Object Identifier (DOI): see top of this page.

[1]Our code and trained model are available at - https://github.com/AccGen99/marl_ipp.

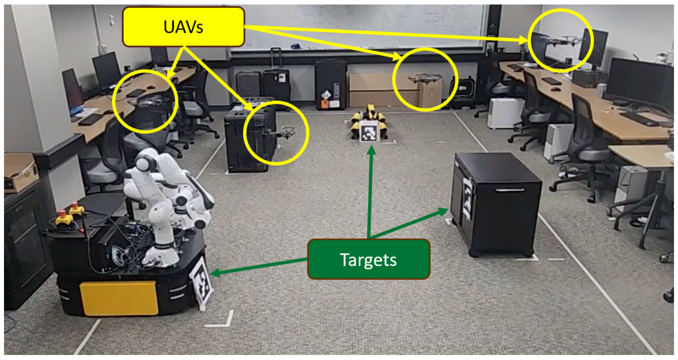

Fig. 1: We implement our approach on Ryze Tello drones in a real-world monitoring scenario. We use Aruco tags as the targets to be discovered. Our approach successfully plans collision-free paths online for maximizing the number of discovered targets while under mission-time constraints. Here, 4 Tello drones and 3 Aruco tags are visible.

budget (e.g., battery capacity or mission time). Our considered problem setting belongs to the family of multi-robot informative path planning (MRIPP) problems. Our 3D environment contains unknown static obstacles. Our multi-robot system consists of unmanned aerial vehicles (UAVs), where each UAV is equipped with two range-constrained modules - a unidirectional RGB-D sensor and a communication module. The challenges considered in this work include - the ability to scale to a larger number of robots deployed in more complex environments, consideration of regions explored by other robots while planning, and avoiding inter-robot and robot-obstacle collisions as the robots operate under communication constraints. Applications of our work include search and rescue missions, reconnaissance for military applications, mapping fruits in an orchard for precision agriculture, and discovering targets of interest in urban environments.

Several approaches have been proposed for the MRIPP problem. Classical approaches [6–10] extend the single-robot planners for multi-robot systems via sequential allocation by planning path for each robot one after another, in a specific sequence. Centralized approaches [2, 11–19] plan over the joint action space of all robots. However, these planners assume availability of global communication and hence are not applicable in our problem setting with a limited communication range. Moreover, as centralized planners plan in the joint action space of all robots, they do not scale well with increasing number of robots. Recently proposed decentralized planners [20–27] decouple the action space - each robot in the multi-robot system plans its own action. However, these approaches do not consider limited communication range [20, 22, 23, 25, 26], inter-robot collision avoidance [22], or presence of unknown static obstacles in the environment [25, 27]. To address these issues, we propose a novel decentralized approach based on deep reinforcement learning. Our approach considers the regions previously explored by other robots during planning, and constrains the planning to each robot's local region. This aids our method in not only avoiding collisions with newly discovered static obstacles, but also in preventing inter-robot collisions. As our approach is decentralized, it can be implemented on larger multi-robot systems deployed within more complex environments unseen during training.

To summarize, we develop a scalable, decentralized, and efficient deep-reinforcement learning-based solution to the MRIPP problem.Our deep reinforcement learning based policy can be implemented on a multi-robot system consisting of a large number of robots in environments not seen during training. The core aspect of our approach is the coordination graph that models the regions of the environment previously explored by other robots. Figure 1 illustrates our approach implemented on a multi-robot system consisting of UAVs in a real-world monitoring scenario. We present the following four contributions:

- Our coordination graph models the regions explored by other robots, enabling the policy to plan actions visiting unexplored regions of the environment.
- Our proposed reward function, aligned with the MRIPP objective, encourages inter-robot communication.
- Our method enables more efficient discovery of targets (66%) compared to state-of-the-art methods (52%) when deployed in previously unseen environments.
- Our learned policy can be deployed over higher number of robots without requiring re-training.

We assess the effectiveness of our approach in an urban monitoring application in a simulator and also perform real-world experiments with Ryze Tello drones.

II. Related Work

Classical approaches [6–10] to the MRIPP problem attempt to extend the single-robot methods to multi-robot planning via sequential allocation [6]. Here, one planner plans the path for each robot one after another in an arbitrary order. These approaches decompose the environment into clusters and plan each robot's path over the clusters [7], use a random sequence order of the robots for sequential allocation [10], or utilize a deep reinforcement learning approach for generating the planning order of the robots [9]. However, these approaches assume infinite communication range and do not consider inter-robot collisions, leading to inapplicability in our problem setting.

Centralized methods [2, 11–19] introduce cooperative behavior by planning in the joint action-space of all robots. Some approaches plan paths that minimize the final uncertainty of a given uncertainty map [11–13]. Other methods decompose the environment into disjoint clusters and allocate one robot per cluster [2, 14, 15, 18]. Other approaches rely on a consensus filter [17], on deep reinforcement learning [16], or attempt to resolve the inter-robot collisions in pre-computed paths [19]. However, these approaches require availability of infinite communication range. Moreover, as they plan in the joint action space of the robots, these methods are not scalable to a large number of robots.

Decentralized planners [20–29] provide scalable solutions by allowing each robot to independently plan their next action. These approaches propose a decentralized variant of Monte-Carlo tree search (MCTS) [24], or utilize consensus filters for encouraging cooperation among the robots [26–28]. Recently, deep reinforcement learning based decentralized planners have been developed that are not only computationally efficient at deployment but also have the capability of generalizing to similar environments not seen during training. These approaches utilize parameter sharing to encourage cooperation among robots [20], employ imitation learning [29], utilize the centralized training and decentralized execution paradigm [21, 25], employ Q-Learning to learn collision avoidance behavior [23], or attempt to utilize attention mechanism for modeling the paths of other robots [22]. However, these approaches operate in 2D environment with known obstacles [29], do not consider inter-robot collisions [22, 23], assume an obstacle-free environment [25, 27], or consider static communication connectivity [28]. A key difference of our approach with the prior works is that each robot models the regions explored by other robots within communication range and plans only within its local region. This encourages visiting unexplored regions, prevents collisions with discovered obstacles by planning in known local regions, and avoids inter-robot collisions due to planning outside of the collision range of other robots. Our results demonstrate that our approach outperforms the state-of-the-art learning and non-learning methods in our considered problem setting and is scalable to larger number of robots deployed in more complex environments not seen during training.

III. Background

A. Problem Setting

In this work, we aim to maximize the number of discovered targets in a 3D environment containing undiscovered static obstacles. Our multi-robot system consists of $N \in \mathbb{Z}^+$ robots constrained to a total resource budget $B \in \mathbb{R}^+$. We model the budget as the sum of the maximum cost of the paths executed by each robot. Hence, each robot $i \in 1, \ldots, N$ receives a budget of $B^i = B/N$. We model the robots as UAVs equipped with a unidirectional range sensor (e.g., RGB-D camera) and a communication module which is single-hop and range-constrained, i.e., robots communicate only when they are within distance $\rho \in \mathbb{R}^+$. To discover the targets, we assume presence of a noiseless classifier while limiting the sensing range to realistically model the reduced prediction confidence with increasing distance to target.

B. Gaussian Processes

Gaussian processes [30] are widely utilized to represent continuous distributions [3, 22, 31] as they enable interpolation

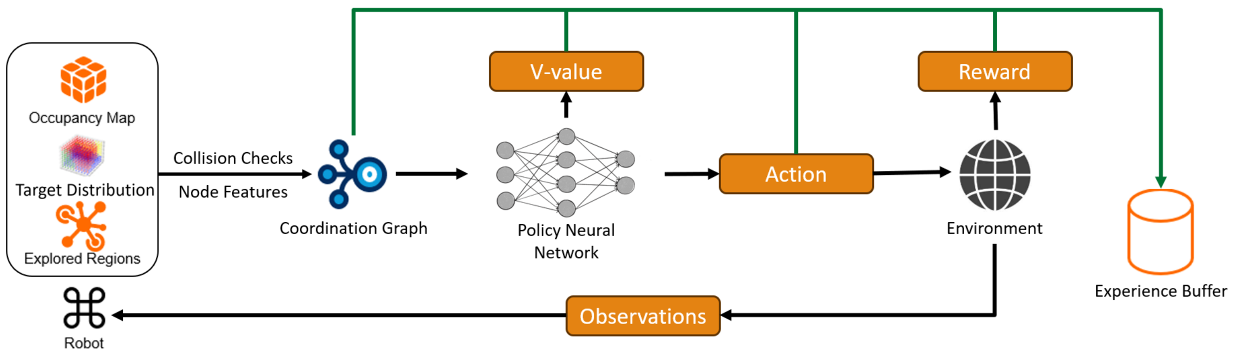

Fig. 2: Overview of our deep reinforcement learning approach for the MRIPP problem. At each time-step, our approach samples collision-free candidate actions in the robot's local region. Our coordination graph associates each candidate action with a utility value, the uncertainty of the utility value, and the exploration features modeling the regions visited by other robots. Our policy network relies on these features to output the robot's state value and the next action to execute, leading to the generation of reward and observations from the environment. Here, the black arrows indicate the robot control loop, green arrows and green boxes are the variables stored in the experience buffer for on-policy training of our policy network.

between discrete measurements. Moreover, in addition to providing predicted values, Gaussian processes have the ability to measure the uncertainty related to the predictions. These uncertainty measures are particularly valuable in our problem setting where understanding the confidence interval around a prediction is crucial for planning subsequent paths.

Given a set of n' features $\mathcal{X}^* \subset \mathcal{X}$ at which a scalar value is to be inferred, a set of n observed feature set $\mathcal{X}' \subset \mathcal{X}$ and the corresponding observed measurements set $\mathcal{Y}$, the mean and covariance of the GP is regressed as:

$$\begin{aligned} u &= \mu(\mathcal{X}^*) + K(\mathcal{X}^*, \mathcal{X}')[K(\mathcal{X}', \mathcal{X}') + \sigma_n^2 I]^{-1}(\mathcal{Y} - \mu(\mathcal{X}')) \\ P &= K(\mathcal{X}^*, \mathcal{X}^*) - K(\mathcal{X}^*, \mathcal{X}')[K(\mathcal{X}', \mathcal{X}') + \sigma_n^2 I]^{-1} \\ &\quad \times K(\mathcal{X}^*, \mathcal{X}')^T \end{aligned}$$

where $K(\cdot)$ is a pre-trained kernel function, σ_n^2 is a hyperparameter describing the measurement noise, and I is the $n \times n$ identity matrix.

IV. OUR APPROACH

In this section, we provide details for each aspect of our proposed deep reinforcement learning based approach to the MRIPP problem. We provide an overview of our approach in Figure 2 for a robot within the multi-robot system.

A. *Environment Representation*

We define the complete action space of the robots as $\mathcal{A}$. We model the disjoint candidate action space for each robot i at timestep t, defined as $\mathcal{A}_t^i \subset \mathcal{A}$, as a set of $j \in \{1, \dots, L\}$ actions $\mathbf{a}_{j,t}^i = (x_{j,t}^i, y_{j,t}^i, z_{j,t}^i, d_{j,t}^i)^\top$ where $|\mathcal{A}_t^i| = L$. Here, we define the robot's 3D coordinates as $x_{j,t}^i, y_{j,t}^i, z_{j,t}^i \in \mathbb{R}$ and the viewing direction for the unidirectional sensor as $d_{j,t}^i \in \mathcal{D}$. We define a set $\mathcal{D}$ to denote possible sensor view directions. At each time-step t each robot i has executed an action $\mathbf{a}_{t-1}^i$. We then plan an action to execute $\mathbf{a}_t^i \in \mathcal{A}_t^i$. Similar to [3], the candidate actions are sampled randomly with a uniform distribution in the robot's C-neighborhood around previous pose $\mathbf{a}_{t-1}^i$. To account for the kinematic constraints of the robot platform, we sample only the feasible actions. Here, C is a constant specifying the extent of the robot's local region. To ensure inter-robot collision avoidance, we constrain the robots to not sample within collision distance $d_c \in \mathbb{R}^+$ of other robots within communication range ρ, and restrict that $d_c < \rho$.

Each robot maintains an occupancy map for collision avoidance with newly discovered obstacles. We initialize the occupancy map voxels as unknown space (1) and update the observed voxels as either free (0) or occupied (2). A voxel is occupied if it contains either a target, or a static obstacle. For robot-obstacle collision avoidance, we perform reachability checks for each candidate action along straight lines.

Execution of an action $\mathbf{a}_{t-1}^i$ by robot i leads to the observation of a certain number of targets at timestep t. To capture the relationship between an action and its corresponding number of observed targets, we define a utility function $u : \mathcal{A} \to \mathbb{R}^+$ for each candidate action. As the utility values for candidate actions are initially unknown, we utilize a Gaussian process [30] to model the function u. The Gaussian process is trained on the utility values $u(\mathbf{a}_t) \in \mathbb{R}^+$ of actions executed till temestep T', i.e. $\mathbf{a}_t \ \forall \ t \in [0, T']$ and is used to regress the utility and uncertainty values of the candidate actions. To stabilize the policy learning, we normalize the observed number of targets by a constant value. The predicted uncertainty values aid our policy network in planning long-horizon paths.

At each timestep t, robot i attempts communication with robot j within communication range ρ. At time-step t, if the Euclidean distance between the robots is less than the maximum communication range $||\mathbf{a}_{t-1}^i - \mathbf{a}_{t-1}^j||_2 \le \rho$, the robots exchange their complete history of the visited waypoints. Each robot maintains a Gaussian process to model the regions explored by other robots as a probability distribution over the robot's workspace. The probability and confidence values queried from the communication Gaussian process over the set of candidate actions aid our policy network in considering the regions explored by other robots while planning the next action.

B. MRIPP Objective

We model the path followed by robot i as a sequence of consecutively executed actions ${\psi^i}_{0:T} = (\mathbf{a}_0^i, \mathbf{a}_1^i, \ldots, \mathbf{a}_T^i)$ where $\mathbf{a}_0^i$ is the initial pose and $\mathbf{a}_T^i$ is the action executed upon depletion of the budget $B^i = B/N$, causing the termination of its mission. In general, the MRIPP problem searches the space of all possible paths $\Psi^{1:N}$ for a set of optimal paths ${\psi^*}_{0:T} \in \Psi^{1:N}$ such that ${\psi^*}_{0:T} = [{\psi^1}_{0:T}, {\psi^2}_{0:T}, \ldots, {\psi^N}_{0:T}]$ to maximize an information-theoretic objective function:

$$ {\psi^*}_{0:T} = \text{argmax} \;\; I(\psi_{0:T}), \;\; \text{s.t.} \; c({\psi^i}_{0:T}) \leq B^i \;\forall\; i \in [1, N]\,, \tag{1} $$

where $I : \Psi \to \mathbb{R}^+$ is the information gained upon executing the trajectory ${\psi^i}_{0:T}$ and $c : {\psi^i}_{0:T} \to \mathbb{R}^+$ maps the path ${\psi^i}_{0:T}$ to its execution cost.

While traversing the path ${\psi^i}_{0:t}$, the robot transitions between two consecutively executed actions over a straight line. Observations are collected at each waypoint in the path and are used to update the utility Gaussian process (u_{util}, P_{util}) and the occupancy map. Upon communication with a nearby robot, the communication Gaussian process (u_{comm}, P_{comm}) is updated with the waypoints visited by the communicating robots. Hence, due to the successive nature of the executed actions in the planned path, we model the MRIPP problem as a sequential decision-making process. Towards the MRIPP objective, we define a function $\zeta : \mathcal{A} \times \Psi^{1:N} \to \mathbb{R}^+$ as the number of new targets observed upon executing an action $\mathbf{a}_t^i$ by robot i after following the path ${\psi^i}_{0:t-1}$.

We define the information obtained by robot i as:

$$ I(\psi_{0:T}) = \sum_{t=1}^{T} \zeta(\mathbf{a}_t^i, {\Psi^{1:N}}_{0:t-1})\,, \tag{2} $$

and we aim to plan ${\psi^*}_{0:T}$ to maximize the information gain. The above formulation models the diminishing information gain by not considering the targets that have been observed during an earlier exploration of the region.

C. Reward Structure

In order to maximize the number of discovered targets, each robot needs to balance exploration of environment (always choosing the action with most uncertainty) with exploitation (always choosing action with maximum informativeness) of the obtained observations. Moreover, inter-robot communication is necessary to keep track of the regions previously explored by other robots. This aids in avoiding planning of sub-optimal actions leading to re-exploration. Inspired by previous works [3, 32], we propose a new reward structure that not only considers the exploration-exploitation trade-off but also encourages inter-robot communication. Upon communication with other robots, information is exchanged about the regions previously explored by the robots. At each time-step t, the robot i has executed the action $\mathbf{a}_{t-1}^i$, collected observations, communicated with the nearby robots, and receives a reward $r_t^i \in \mathbb{R}^+$. The reward function consists of an exploratory term $r_{e,t}^i$, an informative term $r_{u,t}^i$, and a communication term $r_{c,t}^i$ so that:

$$ r_t^i = \alpha r_{e,t}^i + \beta r_{u,t}^i + \gamma r_{c,t}^i \tag{3} $$

where:

$$ \begin{aligned} r_{e,t}^i &= \frac{\text{Tr}(P_{util}^-) - \text{Tr}(P_{util}^+)}{\text{Tr}(P_{util}^-)}\,, \\ r_{c,t}^i &= \frac{\text{Tr}(P_{comm}^-) - \text{Tr}(P_{comm}^+)}{\text{Tr}(P_{comm}^-)}\,, \\ r_{u,t}^i &= \;\zeta(\mathbf{a}_{t-1}, \psi_{0:t-2}) \end{aligned} \tag{4} $$

where the constants α and β balance the exploration-exploitation trade-off and γ rewards inter-robot communication. $\text{Tr}(\cdot)$ is the matrix trace operator. Here, P^- and P^+ indicate the prior and posterior covariance matrices of the Gaussian processes. The reduction in variance of the utility Gaussian process estimates the exploration of the environment due to the robot's own executed actions. Similarly, the variance reduction of the communication Gaussian process estimates the exploration knowledge gained from other robots. The number of new targets observed measures the information gained upon execution of action $\mathbf{a}_t^i$ by robot i. Scaling the reward by $\text{Tr}(P^-)$ stabilizes the actor-critic network training [3, 31].

Our reward generation method ensures that each robot receives the reward reflecting the contribution of it's actions towards the global MRIPP objective -

- Robot i will not receive informative reward $r_{u,t}^i$ for the new targets that have been observed by another robot.
- At each timestep t during training, we utilize a single global utility Gaussian process for all robots. The term $r_{e,t}^i$ is calculated for robot i's action considering no action has been executed by any other robots.
- The communication reward $r_{c,t}^i$ depends on the exploration knowledge gained by robot i from other robots. Each robot has a separate instance of communication Gaussian process to generate this reward component to ensure the reward received depends on the communication performed due to execution of its own action only.

D. Coordination Graph

The MRIPP problem considered in this work requires each robot i in our multi-robot system to reason about the distribution of targets in the environment to optimize the MRIPP objective as described in Equation (1) and the regions explored by other robots for inter-robot collision avoidance. Inspired by the dynamic graph approach for single-robot path planning [3], we propose a novel coordination graph that enables our approach to model the distribution of targets in the robot's local neighborhood, plan actions to visit regions in the environment not explored by other robots, and avoid inter-robot collisions and collisions with newly discovered static obstacles. Our policy neural network relies on the coordination graph to predict the next action to execute.

Each robot rebuilds its coordination graph at every timestep to account for the newly obtained observations. Our coordination graph for a robot i at timestep t is a fully-connected graph

$\mathcal{G}_t^i = (\mathcal{N}_t^i, \mathcal{E}_t^i)$. The node set $\mathcal{N}_t^i$ defines the set of collision-free candidate actions. The edge set $\mathcal{E}_t^i$ defines the collision-free paths from the robot's current pose to each candidate action and each edge in the set is associated with the cost of executing the given candidate action.

We construct the feature matrix $\mathcal{M}_t^i$ for robot i corresponding to its coordination graph as the input to our policy neural network. The features consist of the candidate actions, the utility and uncertainty values of candidate actions regressed from the Gaussian process modeling the utility, and the probability and uncertainty values queried from the communication Gaussian process. The n^{th} row of $\mathcal{M}_t^i$ relates to the n^{th} candidate action of robot i at timestep t:

$$\begin{aligned}\mathbf{M}_t^i(n) = &[\mathbf{a}_{n,t}^i,\, u_{utility}(\mathbf{a}_{n,t}^i),\, P_{utility}(\mathbf{a}_{n,t}^i, \mathbf{a}_{n,t}^i),\\ &u_{comm}(\mathbf{a}_{n,t}^i, t),\, P_{comm}(\mathbf{a}_{n,t}^i, \mathbf{a}_{n,t}^i)]\,,\end{aligned}$$

where $\mathbf{a}_{n,t}^i = [x_{n,t}^i,\, y_{n,t}^i,\, z_{n,t}^i,\, d_{n,t}^i]^\top$, $u_{utility}(\mathbf{a}_{n,t}^i)$ and $P_{utility}(\mathbf{a}_{n,t}^i, \mathbf{a}_{n,t}^i)$ are the regressed utility and uncertainty values for candidate action $\mathbf{a}_{n,t}^i$, and $u_{comm}(\mathbf{a}_{n,t}^i)$ and $P_{comm}([\mathbf{a}_{n,t}^i], [\mathbf{a}_{n,t}^i])$ are the regressed probability and confidence values from the communication Gaussian process modeling the locations of other robots at timestep t.

E. Policy Neural Network

Our coordination graph models each robot's collision free action space and aids the deep reinforcement learning policy in reasoning about the robot's current knowledge of the environment. As the utility Gaussian process can only model the greedy action selection through utility regression, deep reinforcement learning is essential for achieving the balance in short-term exploitation of obtained information and the long-term exploration of the unknown environment.

At each timestep t, each robot i in our multi-robot system utilizes an attention-based neural network [31] to model the planning policy $\pi(\mathcal{G}_t^i, {\psi^i}_{0:t-1}, \tilde{B}^i)$. The planning policy outputs the probability distribution over the candidate actions in the robot's candidate action set $\mathcal{A}_t^i$. The policy relies on the feature matrix $\mathbf{M}_t^i$ of the current coordination graph $\mathcal{G}_t^i$, path executed so far ${\psi^i}_{0:t-1}$ and the remaining budget $\tilde{B}^i$. The network structure consists of an encoder and a decoder module. The encoder models the information distribution obtained from observations and the environment explored so far by learning the dependencies among the candidate actions in $\mathcal{G}_t^i$, forming the context over collected observations. The decoder utilizes the learned context features from the encoder, the planning state, and the budget mask to output the probability distribution over the set of candidate actions $\mathcal{A}_t^i$. The planning state consists of the path executed by the robot so far ${\psi^i}_{0:t-1}$ and the remaining budget $\tilde{B}^i$. The budget mask aids in filtering out candidate actions leading to the violation of the budget constraint. Additionally, the decoder module estimates the value function of the current state. The estimated value, executed actions, coordination graphs, planning states, and rewards generated by actions of the robots throughout the training episode are collected in the experience buffer for on-policy actor-critic reinforcement learning under centralized training and decentralized execution paradigm. In this work, we use proximal policy optimization [33] due to its stability and sample efficiency. During deployment, at each time step and for each robot, we execute the most informative action.

V. RESULTS

A. Setup

Environment. We test our approach in an urban monitoring scenario consisting of buildings and windows. We represent the environment internally as bounded within a scale-agnostic unit cube. Each robot in our multi-robot system maintains an occupancy map for collision avoidance with static obstacles. We initialize the occupancy map as unknown space and update the free space or occupied space based on obtained observations. Our training environment consists of regularly spaced buildings with the windows generated randomly on the buildings. However, our test environments consist of buildings generated at random locations.

Hyperparameters. We tune the hyperparameters of our Gaussian processes in a small representative environment. We use the Matérn $1/2$ kernel function for the Gaussian processes. For the reward structure defined in Equation (1), we choose $\alpha = 20.0$ and $\delta = 0.02$ so that both the exploratory $r_{e,t}^i$ and utility reward $r_{u,t}^i$ terms lie numerically in the range $[0, 1]$. In order to promote MRIPP objective over the inter-robot communication, we use $\gamma = 1.0$ to keep the numerical value of $r_{c,t}^i$ lower than the other terms.

Robot Configuration. We consider each robot as a UAV platform equipped with an RGB-D camera having $90°$ field of view. We model the reduction in the confidence of target identification with increasing distance by limiting the camera sensing range to 24% of the environment size. The UAVs can communicate at a maximum communication distance $\rho = 0.3$. The sensor viewpoint set $\mathcal{D}$ is discretized as $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ radians. However, our approach supports extension to finer discretizations by extending the set $\mathcal{D}$.

Training. We generate multiple training episodes to populate our experience buffer. Each training episode consists of a multi-UAV system with a total budget B. Our policy is trained in a structured environment and then transferred to a randomized environment for testing. While we fix the number of buildings during training, the number of windows is varied in $[200, 250]$. The start action for each robot is $\mathbf{a}_0 = (0.0, 0.0, 0.0, \frac{\pi}{2})$. We set $L = 80$ for each robot's coordination graph. Since we normalize the internal environment representation, our budget value B is unitless. For each training episode, B is a randomly generated real value in the range $[7.0, 9.0]$. Each episode is constrained to a maximum of 256 timesteps. To speed-up the training process, we run 36 parallel environment instances and train our policy network over 8 epochs with a batch size of 1024. We utilize Adam optimizer with a learning rate of 10^{-4}, decaying by a factor of 0.96 after every 512 optimization steps. The policy gradient epsilon-clip parameter is set to 0.2. We train our policy network on a computing cluster equipped with Intel(R) Xeon(R) CPUs @ 3.60GHz and one NVIDIA A30 Tensor Core GPU. We require $\sim 120,000$ environment interactions for our policy to converge. All our tests are conducted on the same compute cluster.

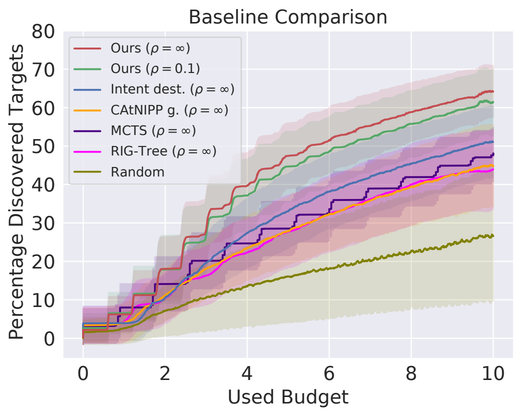

Fig. 3: Comparison of our approach with other baselines in an urban environment. Our performance metric is the percentage of targets discovered during the episode. The solid lines represent the mean values across 250 trials, while the shaded areas denote the standard deviations.

B. Baseline Comparison

In this section, we compare the performance of our approach against state-of-the-art learning and non-learning baselines. We utilize fixed random seeds to generate 25 test environments. We enable global communications for the baselines and present the performance of our approach with both global communication ($\rho = \infty$) and restricted communication ($\rho = 0.1$). We deploy $N = 3$ UAVs with a total budget of $B = 10.00$ and run 10 trials corresponding to each randomly generated environment, leading to a total of 250 tests. Our baselines include: (i) Intent with destination modeling [22] as a zero-shot greedy policy (Intent dest.), (ii) CAtNIPP [31] with a zero-shot policy (CAtNIPP g.), (iii) non-learning Monte Carlo Tree Search [34] (MCTS), (iv) non-learning Rapidly exploring random Information Gathering Trees [35] (RIG-Tree), and (v) a random policy (random agent). As MCTS, and RIG-Tree are single robot planners, we extend them to multi-robot planning via sequential allocation [6]. We tune all the baselines for best performance in our problem setting. We consider the metric of percentage of targets discovered, as well as provide the average planning time per step. Additionally, as modification of these approaches to account for collision-free path planning is non-trivial, we allow the UAVs for each planner to ignore obstacle-UAV and inter-robot collisions, ensuring a fair comparison within this set of experiments.

Our results shown in Figure 3 and Table I indicate that our method outperforms the considered baselines in terms of the number of discovered targets. Note that introduction of communication constraint leads to a drop in the performance of our approach, however we still outperform the considered baselines operating with global communications. This could be attributed to our coordination graph explicitly representing the regions explored by other robots, leading to planning actions visiting unexplored regions that provide performance improvement in our approach over the baselines that do not model the regions visited by other robots. Note that deep reinforcement learning based methods are significantly more time-efficient than non-learning approaches, justifying their use over non-learning methods for real-time applications.

TABLE I: Results of our approach compared with other learning and non-learning baselines in an urban monitoring scenario.

Baseline	% targets	Time (s)
Our approach ($\rho = \infty$)	$\mathbf{66.00 \pm 5.59}$	$\mathbf{1.58}$
Our approach ($\rho = 0.1$)	$\mathbf{62.16 \pm 8.25}$	$\mathbf{1.58}$
Intent dest. ($\rho = \infty$)	52.31 ± 9.29	25.39
CAtNIPP g. ($\rho = \infty$)	45.31 ± 10.59	3.84
MCTS ($\rho = \infty$)	48.50 ± 7.81	153.49
RIG-Tree ($\rho = \infty$)	47.38 ± 10.43	60.46
Random agent	31.83 ± 15.23	0.06

TABLE II: Ablation study for modeling of explored regions.

Approach	% targets
With modeling explored regions	66.00 ± 5.59
Without modeling explored regions	61.97 ± 7.34

C. Ablation Studies

We study the impact of our communication Gaussian process and our new reward structure on the performance of our approach via the metric of percentage discovered targets.

Communication Gaussian Process. To evaluate the impact of modeling the unexplored regions of the environment on the performance of our approach, we compare our approach trained with and without the communication Gaussian process. Our results in Table II show that the performance improves when the communication Gaussian process is included. Furthermore, an unpaired t-test conducted between the two groups ($n = 250$ each) yielded a p-value of 3.91×10^{-9}, indicating a statistically significant difference well below the conventional threshold of 0.05. Hence, we conclude that our policy neural network learns to reason about the unexplored regions during planning.

Communication Reward. To evaluate the impact of the communication reward term $r^i_{c,t}$ in Equation (4) on the performance of our approach, we compare the performance of our policies trained with $\gamma = 0.0$ and $\gamma = 1.0$. Our results in Table III show that the performance improves upon inclusion of the communication reward. Again, the difference between the two groups of 250 tests each is statistically significant, with a p-value of 5.11×10^{-6}, significantly less than 0.05, providing strong evidence against the null hypothesis. Hence, our new reward structure promotes inter-robot communication and leads to improved performance.

D. Scalability

We compare the ability of our approach to scale to larger environments and more number of robots N in the multi-robot system with other approaches. Our policy learned in the small training environment with $N = 3$ robots is evaluated in larger environments and varying number of robots not seen during training. We present results for test environments that are approximately $3\times, 8\times$, and $16\times$ larger than the training

TABLE III: Ablation study for reward structure.

Approach	% targets
With communication reward	66.00 ± 5.59
Without communication reward	62.92 ± 7.01

TABLE IV: Comparison of our deep reinforcement learning-based approach against baselines in an urban environment.

Approach	% targets
Our Approach	66.46 ± 11.08
Random Planner	28.94 ± 8.70

environment. We consider $N \in (16,\ 32,\ 48,\ 64)$ and conduct 100 tests in each environment for every N.

We compare the performance of our approach, deployed with communication distance $\rho \in (0.2, 0.15, 0.12)$ in $3\times, 8\times, 16\times$ environment respectively, with (i) CAtNIPP [31] with global observability and (ii) a random policy. We train CAtNIPP in the training environment described in Section V-A. We do not consider the intent baseline due to its compute intensive nature for large multi-robot systems. We do not evaluate the non-learning methods, as the sequential allocation based time-intensive nature leads to infeasible computation times for large multi-robot systems.

Figure 4 demonstrates the results for this set of experiments. Our approach strongly outperforms the considered baselines in $3\times$ environment, slightly outperforms in $8\times$ environment, and is outperformed in $16\times$ environment. The success of our planner in $3\times$ and $8\times$ environments can be attributed to the modeling of unexplored regions, leading to planning of more informative paths by robots in our multi-robot system as compared to other approaches. However, in the $16\times$ environment, the instances of communication with robots located further away drastically reduces, causing our approach to be outperformed by the CAtNIPP planner with global observability. Future work will explore more complex communication paradigms to mitigate this issue.

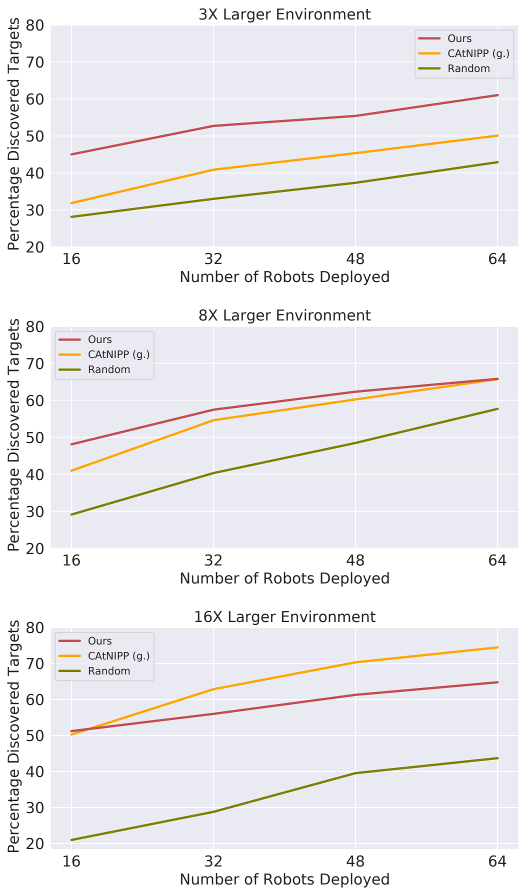

Fig. 4: Our approach outperforms other baselines in terms of percentage discovered targets in a $3\times$, $8\times$, and $16\times$ larger environments. The x-axis indicates the number of robots in the multi-robot system during the test. The solid lines represent the mean values across 100 trials.

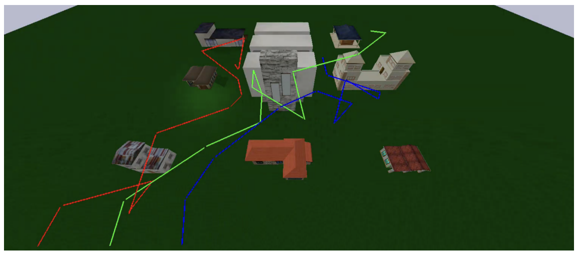

Fig. 5: Our approach implemented in an urban simulation environment. We place 3 UAVs and 161 targets in the environment and trace each UAV's path with colored tracelines.

E. Simulation

We demonstrate the applicability of our deep reinforcement learning approach in an urban monitoring scenario. We use the gym-PyBullet-drones [36] simulator to accurately model UAV physics. Our simulation environment is built using the Houses3K dataset [37] and is bounded by a $60\,\mathrm{m} \times 60\,\mathrm{m} \times 30\,\mathrm{m}$ cuboid as shown in Figure 5. We assume perfect localization and use ground truth target discovery. The 3 UAVs move at a maximum speed of $1\,\mathrm{m/s}$.

We compare the performance of our approach with a random planner that reflects the performance lower bound. We do not implement other baselines considered in Section V-B as modifying these approaches for avoiding inter-robot collisions and consideration of the presence of unknown obstacles in the environment is a non-trivial task. Here, our evaluation metric is the percentage of windows discovered by the robots. To ensure every discovered target is counted only once, we record the coordinates of discovered targets. Our results are reported for missions with a budget of 7.0 units in Table IV. Our approach outperforms the random planner.

F. Implementation

We demonstrate the real-world applicability of our method on a multi-robot system for target discovery as illustrated in Figure 1. We carried out experiments on 4 Ryze Tello drones in a $7.62 \times 3.25 \times 2.4$ m^3 arena containing randomly placed obstacles and 6 Aruco tags as targets.

VI. Conclusion

We present a novel deep reinforcement learning approach for the MRIPP problem in an unknown 3D environment. Our coordination graph-based approach models the unexplored regions of the environment for efficient target discovery. We present experimental results to support that: (i) our coordination graph encourages exploration of unknown regions of the environment, (ii) our reward function encourages inter-robot communication, (iii) our approach outperforms the state-of-the-art baselines in environments unseen during training, and (iv) our learned policy scales to larger multi-robot systems and more complex environments. We evaluate the performance of our approach in a UAV-based urban mapping scenario in a simulator, as well as conduct real robot experiments to demonstrate the practical applicability. Future research directions include extension to multi-robot pathfinding, task allocation, accounting for localization errors, and cooperative communication.

Acknowledgment

This material is based upon work supported in part by the DEVCOM Army Research Laboratory under cooperative agreement W911NF2520170.

References

[1] J. Berger and N. Lo, "An innovative multi-agent search-and-rescue path planning approach," *Computers & Operations Research*, vol. 53, pp. 24–31, 2015.

[2] S. Yoon and C. Qiao, "Cooperative search and survey using autonomous underwater vehicles (auvs)," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 364–379, 2010.

[3] A. Vashisth, J. Ruckin, F. Magistri, C. Stachniss, and M. Popovic, "Deep reinforcement learning with dynamic graphs for adaptive informative path planning," *IEEE Robotics and Automation Letters (RA-L)*, 2024.

[4] Y. U. Cao, A. B. Kahng, and A. S. Fukunaga, "Cooperative mobile robotics: Antecedents and directions," *Robot colonies*, pp. 7–27, 1997.

[5] J. Su, X. Zhu, S. Li, and W.-H. Chen, "AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture," *Neurocomputing*, vol. 518, pp. 242–270, 2022.

[6] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, "Efficient planning of informative paths for multiple robots," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007, pp. 2204–2211.

[7] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research (JAIR)*, vol. 34, pp. 707–755, 2009.

[8] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

[9] S. Y. Luis, D. Shutin, J. M. Gómez, D. G. Reina, and S. T. Marín, "Deep reinforcement multi-agent learning framework for information gathering with local gaussian processes for water monitoring," *arXiv preprint arXiv:2401.04631*, 2024.

[10] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. IEEE, 2010, pp. 4457–4462.

[11] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, 2008.

[12] A. Dutta, A. Ghosh, and O. P. Kreidl, "Multi-robot informative path planning with continuous connectivity constraints," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. IEEE, 2019.

[13] Y. Wei and R. Zheng, "Multi-robot path planning for mobile sensing through deep reinforcement learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[14] G. A. Di Caro and A. W. Z. Yousaf, "Multi-robot informative path planning using a leader-follower architecture," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*. IEEE, 2021.

[15] D. S. Diop, S. Y. Luis, M. P. Esteve, S. L. T. Marín, and D. G. Reina, "Decoupling patrolling tasks for water quality monitoring: A multi-agent deep reinforcement learning approach," *IEEE Access*, 2024.

[16] A. M. Barrionuevo, S. Y. Luis, D. G. Reina, and S. L. T. Marín, "Informative deep reinforcement path planning for heterogeneous autonomous surface vehicles in large water resources," *IEEE Access*, 2024.

[17] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, 2014.

[18] N. Cao, K. H. Low, and J. M. Dolan, "Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms," *arXiv preprint arXiv:1302.0723*, 2013.

[19] R. Cui, B. Gao, and J. Guo, "Pareto-optimal coordination of multiple robots with safety guarantees," *Autonomous Robots*, vol. 32, 2012.

[20] A. Viseras and R. Garcia, "Deepig: Multi-robot information gathering with deep reinforcement learning," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 3, pp. 3059–3066, 2019.

[21] J. Westheider, J. Rückin, and M. Popović, "Multi-uav adaptive path planning using deep reinforcement learning," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

[22] T. Yang, Y. Cao, and G. Sartoretti, "Intent-based deep reinforcement learning for multi-agent informative path planning," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 71–77.

[23] S. Yanes Luis, M. Perales Esteve, D. Gutiérrez Reina, and S. Toral Marín, "Deep reinforcement learning applied to multi-agent informative path planning in environmental missions," in *Mobile Robot: Motion Control and Path Planning*. Springer, 2023, pp. 31–61.

[24] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-mcts: Decentralized planning for multi-robot active perception," *Int. Journal of Robotics Research (IJRR)*, vol. 38, no. 2-3, pp. 316–337, 2019.

[25] F. Venturini, F. Mason, F. Pase, F. Chiariotti, A. Testolin, A. Zanella, and M. Zorzi, "Distributed reinforcement learning for flexible uav swarm control with transfer learning capabilities," in *Proceedings of the 6th ACM workshop on micro aerial vehicle networks, systems, and applications*, 2020, pp. 1–6.

[26] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2015.

[27] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *Int. Journal of Robotics Research (IJRR)*, vol. 31, no. 10, 2012.

[28] A. Asgharivaskasi, F. Girke, and N. Atanasov, "Riemannian optimization for active mapping with robot teams," *IEEE Transactions on Robotics*, vol. 41, pp. 1077–1097, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269448652

[29] M. Tzes, N. Bousias, E. Chatzipantazis, and G. J. Pappas, "Graph neural networks for multi-robot active information acquisition," *arXiv preprint arXiv:2209.12091*, 2022.

[30] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[31] Y. Cao, Y. Wang, A. Vashisth, H. Fan, and G. A. Sartoretti, "Catnipp: Context-aware attention-based network for informative path planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1928–1937.

[32] D. M. S. Tan, Y. Ma, J. Liang, Y. C. Chng, Y. Cao, and G. Sartoretti, "Ir 2: Implicit rendezvous for robotic exploration teams under sparse intermittent connectivity," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 13 245–13 252.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*.

[34] J. Ott, E. Balaban, and M. J. Kochenderfer, "Sequential bayesian optimization for adaptive informative path planning with multimodal sensing," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2023.

[35] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. Journal of Robotics Research (IJRR)*, vol. 33, no. 9, pp. 1271–1287, 2014.

[36] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 7512–7519.

[37] D. Peralta, J. Casimiro, A. M. Nilles, J. A. Aguilar, R. Atienza, and R. Cajote, "Next-best view policy for 3d reconstruction," in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 558–573.